

METHOD AND DEVICE FOR DATA COMPRESSION

Patent number: JP9069785

Publication date: 1997-03-11

Inventor: TODA ATSUKO

Applicant: TOSHIBA CORP

Classification:

- International: H03M7/42

- european:

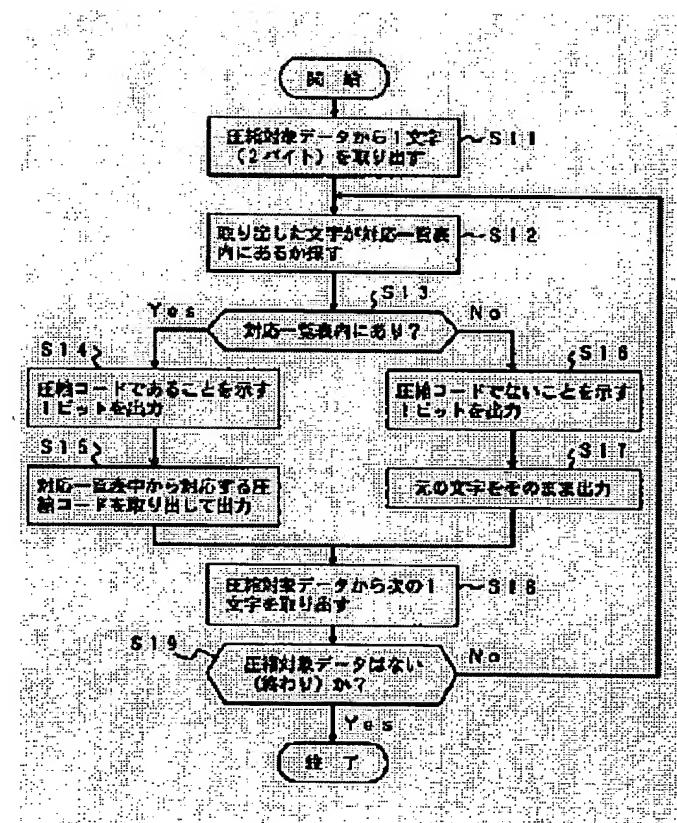
Application number: JP19950222155 19950830

Priority number(s):

Abstract of JP9069785

PROBLEM TO BE SOLVED: To realize effective data compression in the case of taking Japanese- language text data as the compression object.

SOLUTION: A correspondence list where code data of specific characters and their compressed code data correspond to each other is prepared, and it is retrieved whether character code data of one character taken out from compression object data exists in the correspondence list or not (S11 and S12). If it exists in the list as the result (S13), its compressed code data is read from the correspondence list and is outputted (S14 and S15); but if it doesn't exist there (S13), data is outputted as it is (S16 and S17).



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-69785

(43)公開日 平成9年(1997)3月11日

(51)Int.Cl.⁶
H 03M 7/42

識別記号
H 03M 7/42

府内整理番号
9382-5K

F I
H 03M 7/42

技術表示箇所

審査請求 未請求 請求項の数4 OL (全5頁)

(21)出願番号 特願平7-222155

(22)出願日 平成7年(1995)8月30日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 戸田 亜津子

愛知県瀬戸市穴田町991番地 株式会社東
芝愛知工場内

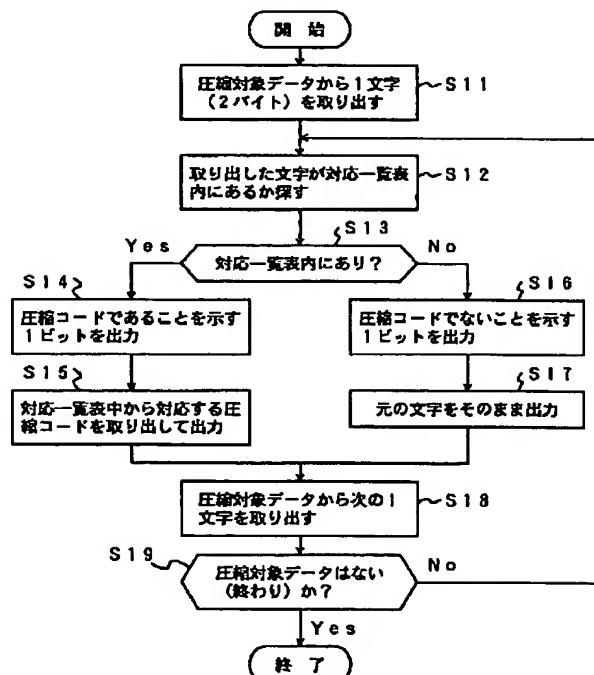
(74)代理人 弁理士 鈴江 武彦

(54)【発明の名称】 データ圧縮方法及びデータ圧縮装置

(57)【要約】

【課題】日本語のテキストデータを圧縮対象とした場合に効果的なデータ圧縮を行う。

【解決手段】特定文字のコードデータとその圧縮コードデータとを対応付けた対応一覧表を用意しておき、圧縮対象データから取り出した1文字分の文字コードデータがその対応一覧表に存在する否かを検索する(S11, S12)。その結果、当該データが対応一覧表に存在する場合には(S13)、その圧縮コードデータを対応一覧表から読み出して出力し(S14, S15)、存在しない場合には(S13)、当該データをそのまま出力する(S16, S17)。



【特許請求の範囲】

【請求項1】 特定文字のコードデータとその圧縮コードデータとを対応付けた対応一覧表を有し、圧縮対象データが上記対応一覧表に存在するか否かを判断し、当該圧縮対象データが上記対応一覧表に存在する場合には、その文字コードに対応する圧縮コードデータを上記対応一覧表より取り出して出力し、当該圧縮対象データが上記対応一覧表に存在しない場合には、その文字コードデータをそのまま出力するようにしたことを特徴とするデータ圧縮方法。

【請求項2】 出力データが圧縮されたデータか否かを識別するための情報をその出力データに先立って出力するようにしたことを特徴とする請求項1記載のデータ圧縮方法。

【請求項3】 特定文字のコードデータとその圧縮コードデータとを対応付けた対応一覧表を記憶するための記憶手段と、

圧縮対象データが上記記憶手段に記憶された対応一覧表に存在するか否かを判断する判断手段と、この判断手段により当該圧縮対象データが上記対応一覧表に存在すると判断された場合には、その文字コードに対応する圧縮コードデータを上記対応一覧表より取り出して出力し、当該圧縮対象データが上記対応一覧表に存在しない場合には、その文字コードデータをそのまま出力する出力手段とを具備したことを特徴とするデータ圧縮装置。

【請求項4】 上記出力手段は、出力データが圧縮されたデータか否かを識別するための情報をその出力データに先立って出力することを特徴とする請求項3記載のデータ圧縮装置。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】本発明は、デジタルデータを扱うコンピュータなどのデータ処理装置に用いられるデータ圧縮方法及びデータ圧縮装置に関する。

【0002】

【従来の技術】従来、デジタルデータを扱うコンピュータなどのデータ処理装置では、例えばハフマン符号化式のように、1バイト単位のデータの出現頻度の違いを利用して、デジタルデータを圧縮する装置があった。

【0003】この他にも、コンピュータが扱うデータ圧縮には様々な方法が存在するが、これらの基本的な方法は外国で考え出されたものであり、日本語の特徴を利用したものではなかった。

【0004】

【発明が解決しようとする課題】ところで、日本語の文章の中には、平仮名、片仮名、数字、アルファベット、句読点といった文字が多く見られる。このような文字を対象としてデータ圧縮を考えた場合、上述したような方

法では単に出現頻度の違いだけでデータ圧縮を行う方法であるため、効果的なデータ圧縮を行うことはできなかった。

【0005】また、圧縮対象データの最小単位は1バイトであることから、2バイトを基本とする日本語のテキストデータには最適とはいえない。本発明は上記のような点に鑑みなされたもので、日本語のテキストデータを圧縮対象とした場合に効果的なデータ圧縮を行うことのできるデータ圧縮方法及びデータ圧縮装置を提供することを目的とする。

【0006】

【課題を解決するための手段】本発明は、特定文字のコードデータとその圧縮コードデータとを対応付けた対応一覧表を用意しておき、圧縮対象データが上記対応一覧表に存在する場合にその文字コードデータに対応する圧縮コードデータを出し、上記対応一覧表に存在しない場合にその文字コードデータをそのまま出力するようにしたものである。

【0007】このような構成によれば、例えば平仮名、片仮名、数字、アルファベット、句読点といった日本語のテキストデータの中で出現頻度の高い文字を対象に、その文字コードデータとその圧縮コードデータを対応一覧表に登録しておけば、日本語の特徴を生かしたデータ圧縮を行うことができる。

【0008】

【発明の実施の形態】以下、図面を参照して本発明の一実施形態を説明する。図1は本発明の一実施形態に係るデータ圧縮装置の構成を示すブロック図である。本装置は、デジタル化された日本語テキストデータに対して圧縮処理を行い、その結果として入力データ数よりも、出力データ数が少なくなるように変形することを目的としたシステムであり、CPU11、メモリ12、補助記憶装置13を主として構成される。

【0009】CPU11は、本装置全体の制御を行うものであり、ここではデータ圧縮処理を実行する。メモリ12は、例えばROMあるいはRAMからなり、ここでは対応一覧表12aを記憶している。対応一覧表12aは、平仮名、片仮名、数字、アルファベット、句読点といった日本語のテキストデータの中で出現頻度の高い文字を対象に、その文字コードデータと圧縮コードデータとを対応付けている。

【0010】補助記憶装置13は、例えばフロッピーディスク装置(FDD)あるいはハードディスク装置(HDD)からなり、ここでは圧縮対象データ(圧縮前のデータ)と圧縮後のデータを格納する。

【0011】図2は同実施形態における対応一覧表12aの内容を示す図である。対応一覧表12aには、平仮名、片仮名、数字、アルファベット、句読点といった日本語のテキストデータの中で出現頻度の高い文字のコードデータとそれに対応する圧縮コードデータが格納され

ている。この例では、文字コードデータは16ビット、圧縮コードは8ビット(16進)で表現されている。

【0012】次に、同実施形態の動作を説明する。図3は同実施形態におけるデータ圧縮処理の動作を示すフローチャートである。CPU11は、まず、補助記憶装置13などに保持されている圧縮対象データの中から1文字分(2バイト)の文字コードデータを取り出す(ステップS11)。

【0013】次に、CPU11はメモリ12に記憶された対応一覧表12aをアクセスし、上記取り出した文字コードデータが対応一覧表12a内にあるか否か検索する(ステップS12)。

【0014】ここで、当該文字コードデータが対応一覧表12a内に存在した場合には(ステップS13のYes)、その圧縮コードデータを出力することになる。その際に、CPU11は圧縮コードデータであることを示す1ビットの情報「1」を出力し(ステップS14)、次に対応一覧表12aの中から当該文字コードデータに対応する圧縮コードデータを読み出し、これを出力する(ステップS15)。

【0015】一方、当該文字コードデータが対応一覧表12a内に存在しなかった場合には(ステップS13のNo)、そのまま圧縮せずに出力することになる。その際に、CPU11は圧縮コードデータでないことを示す1ビットの情報「0」を出力し(ステップS16)、次に対応一覧表12aの中から当該文字コードデータに対応する圧縮コードデータを読み出し、これを出力する(ステップS17)。

【0016】以後、同様にして、圧縮対象データの中から順次文字コードデータを取り出し、これを対応一覧表12aと照らし合わせながら適宜圧縮して出力する(ステップS18, S19)。これにより、対応一覧表12aに存在する文字コードデータについては圧縮して出力することができる。

【0017】ここで、具体的を挙げて上記処理を説明する。例えばステップS11またはS18で「ア」という片仮名の文字コードデータが圧縮対象データの中から取り出されたと仮定すると、処理はS12→S13→S14→S15と進む。

【0018】圧縮コードを示す1ビットの情報を「1」とすると、ステップS14で「1」が出力され、ステップS15で「92」(16進)が出力される。このステップS14、S15で出力されたデータを2進数で表現すると、図4(a)に示すように「110010010」となる。

【0019】また、例えばステップS11またはS18で「亞」という漢字の文字コードデータが圧縮対象データ

の中から取り出されたと仮定すると、処理はS12→S13→S16→S17と進む。

【0020】圧縮コードでないことを示す1ビットの情報を「0」とすると、ステップS16で「0」が出力され、ステップS17で「3021」(16進)が出力される。このステップS16、S17で出力されたデータを2進数で表現すると、図4(b)に示すように「00011000000100001」となる。

【0021】このように、日本語の文字単位(16ビット)で、出現頻度の高い文字のコードデータを予め対応一覧表12aに登録しておき、圧縮対象データが対応一覧表12aに存在する場合に、これを8ビットに置き換えて出力する。これにより、例えば図5に示すように「コンピュータで扱うデータ」といった文字列データがあった場合、圧縮前に192ビットあったものが、圧縮後には116ビットに減らして出力することができる。

【0022】

【発明の効果】以上のように本発明によれば、対応一覧表に存在する文字コードデータを圧縮コードデータに置き換えて出力するようにしたため、例えば平仮名、片仮名、数字、アルファベット、句読点といった日本語のテキストデータの中で出現頻度の高い文字を対象に、その文字コードデータと圧縮コードデータを対応一覧表に登録しておけば、日本語の特徴を生かしたデータ圧縮を行うことができる。

【0023】この場合、基本的にLZ法などの適応型圧縮法と異なり、以前に出現したデータを利用しないので、対象データが短くても圧縮することができる。また、静的ハフマン符号化と異なり、対応一覧表を出力しないので、圧縮率を高められる。さらに、ハフマン符号化と異なり、データの出現頻度を算出する処理が不要であるので処理速度を高速化できる等の効果がある。

【図面の簡単な説明】

【図1】本発明の一実施形態に係るデータ圧縮装置の構成を示すブロック図。

【図2】同実施形態における対応一覧表の内容を示す図。

【図3】同実施形態におけるデータ圧縮処理の動作を示すフローチャート。

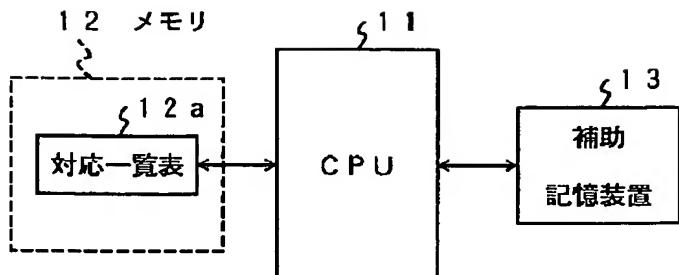
【図4】同実施形態における圧縮コードの形を説明するための図。

【図5】同実施形態におけるデータ圧縮結果を示す図。

【符号の説明】

1 1…CPU、
1 2…メモリ、
1 2 a…対応一覧表、
1 3…補助記憶装置。

【図1】



【図2】

数学	0 (2 3 3 0	0 0
	1 (2 3 3 1	0 1
	:	:	:
	9 (2 3 3 9	0 9
	A (2 3 4 1	0 A
	:	:	:
	Z (2 3 5 A	2 3
	a (2 3 6 1	2 4
	:	:	:
	z (2 3 7 A	3 D
ひらがな	あ (2 4 2 1	3 E
	あ (2 4 2 2	3 F
	い (2 4 2 3	4 0
	い (2 4 2 4	4 1
	:	:	:
カタカナ	ん (2 4 7 3	9 0
	ア (2 5 2 1	9 1
	ア (2 5 2 2	9 2
	:	:	:
その他	ケ (2 5 T 6	E 6
	空白 (2 1 2 1	E 7
	、 (2 1 2 2	E 8
	。 (2 1 2 3	E 9
		圧縮コード (8ビット)	出現頻度の高い文字データ (16ビット)

【図4】

(a)	<table border="1"> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> </table>	1	1	0	0	1	0	0	1	0	「ア」を示す圧縮コード「9 2」(16進)						
1	1	0	0	1	0	0	1	0									
—— 圧縮コードであることを示すデータ																	
(b)	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> </table>	0	0	0	1	1	0	0	0	0	0	1	0	0	0	1	「星」を示す文字コード「3 0 2 1」(16進)
0	0	0	1	1	0	0	0	0	0	1	0	0	0	1			
—— 圧縮コードでないことを示すデータ																	

【図5】

(a) 圧縮前データ	<table border="1"> <tr><td>ヨ</td><td>ン</td><td>ビ</td><td>ュ</td><td>ー</td><td>フ</td><td>で</td><td>絞</td><td>う</td><td>デ</td><td>ー</td><td>フ</td></tr> <tr><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td></tr> </table>	ヨ	ン	ビ	ュ	ー	フ	で	絞	う	デ	ー	フ	16	16	16	16	16	16	16	16	16	16	16	16	計 192 ビット
ヨ	ン	ビ	ュ	ー	フ	で	絞	う	デ	ー	フ															
16	16	16	16	16	16	16	16	16	16	16	16															
(b) 圧縮後データ	<table border="1"> <tr><td>ヨ</td><td>ン</td><td>ビ</td><td>ュ</td><td>ー</td><td>フ</td><td>で</td><td>絞</td><td>う</td><td>デ</td><td>ー</td><td>フ</td></tr> <tr><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td><td>16</td></tr> </table>	ヨ	ン	ビ	ュ	ー	フ	で	絞	う	デ	ー	フ	16	16	16	16	16	16	16	16	16	16	16	16	計 116 ビット
ヨ	ン	ビ	ュ	ー	フ	で	絞	う	デ	ー	フ															
16	16	16	16	16	16	16	16	16	16	16	16															

【図3】

